1 Confidentiality

Confidentiality can be achieved through Symmetric Key Encryption (SE). In the following sections we will study about symmetric key encryption schemes and their properties.

1.1 Symmetric Key Encryption (SE)

Symmetric Key Encryption consists of the following triplet of algorithms:

- Key Generation
- Encryption
- Decryption

All the above run on an input of a security parameter ‘k’

1. Key Generation: A randomized algorithm (that uses “coin flipping”), which produces a random key of length k. A different key is generated every time the algorithm is run. The input, in this case, for the algorithm is null. The randomized output is represented with a dollar symbol ($):

   (a) Input: Null
   (b) $K \leftarrow \text{KeyGen}()$

2. Encryption: A randomized algorithm that takes a plaintext $m$ and the key $K$ as an input, and produces the cipher text $c$ as the output.

   (a) Input: $m \in M, K$ where $M$ is the message space and $k$ is the key generated by the $\text{KeyGen}()$ algorithm.
   (b) $c \leftarrow \text{Enc}(m, k)$ (otherwise $\phi$ if $m \notin M$) $\forall c \in C$, where $C$ is the ciphertext space

3. Decryption: This is a deterministic algorithm.
(a) Input: $c \in C, K$ where $C$ is the Cipher text space and $K$ is a key generated by the $KeyGen()$ algorithm.

(b) Output: $m \leftarrow Dec(c, k) \quad \forall c \in C$ and a key $K$.

1.2 Properties of Symmetric Key Encryption

1.2.1 Correctness Property

It ensures that if message is encrypted using a key, then the same message is returned as the output when the cipher text is decrypted i.e. If $c \leftarrow Enc(m, K)$ then $m \leftarrow Dec(c, K) \quad \forall m \in M, K \in KeySpace$

1.2.2 Security Property

We study the properties an SE possesses in order to be securely used for achieving confidentiality. In other words, we study various notions of security of an SE.

The first property that we study is called perfect or unconditional or information-theoretic secrecy. In achieving perfect secrecy, we consider an adversary who has an unbounded computational power (i.e., it can take infinite amount of computing power to break an SE).

Definition 1 (Perfectly Secure Encryption.) A symmetric encryption is said to be perfectly secure if,

$$Pr(m = m') = Pr(m = m'|c = c') \quad \forall m \in M, c \in C, c' = Enc(m', K)$$

Intuitively, this means that in a perfectly secure encryption, the knowledge of cipher text is as good as no knowledge of it.

Definition 2 (Perfectly Secure Encryption.) A symmetric encryption is said to be perfectly secure if,

$$Pr(Enc(m1, k) = c) = Pr(Enc(m2, k) = c) \quad \forall m1, m2 \in M, c \in C$$

Intuitively, this means that all the messages are equally likely to be the plaintext corresponding to a given ciphertext.

The two definitions can be shown to be equivalent. In other words, Definition 1 $\iff$ Definition 2
**One Time Pad (OTP): A Perfectly Secure Encryption Scheme.** This encryption scheme provides perfectly secrecy. The message space and key space are of same size. It consists of:

- **Key Generation Algorithm:** A randomized algorithm that outputs a random key.
  - Input: Null
  - \( K \leftarrow \text{KeyGen}() \)

- **Encryption:** The cipher text is obtained by XOR operation of message \( m \) with key \( K \) i.e.
  - Input: \( m \in \{0, 1\}^k \)
  - Output: \( c = m \oplus K \)

- **Decryption:** The message \( m \) is retrieved by XOR operation of cipher text \( c \) with key \( K \) i.e.
  \[ m = c \oplus K \]

**Theorem 1** One Time Pad is perfectly secure

**Proof.** In an OTP, \( C = m \oplus K \) where \( K = \{0, 1\}^l \) and \( l \) is the length of the key.

\[
\forall m \in M \\
Pr(Enc(m, k) = c) \\
= Pr(m \oplus k = c) \\
= Pr(k = m \oplus c) \\
= \frac{1}{2^l}
\]

This means that each \( m \in M \) is equally likely to be the plaintext corresponding to a given ciphertext. Therefore, from Definition 2 (under Perfectly Secure Encryption), we can conclude that OTP is perfectly secure.

**Example to illustrate perfect secrecy of OTP:** Let \( l = 2 \).

Given that the ciphertext \( c = '10' \) Since, \( c = m \oplus K \)

\[
m = c \oplus K \\
10 = 10 \oplus 00 \\
00 = 10 \oplus 10 \\
11 = 10 \oplus 01 \\
01 = 10 \oplus 11
\]

\[
Pr(Enc(m, k) = c) = \frac{1}{2^2} = \frac{1}{4}
\]

2-3
It can be seen from the above example that a single cipher text ‘10’ is generated for 4 different possible combinations $K$ and $m$. So the probability of figuring out the message is $\frac{1}{4}$.

**Theorem 2** *(Impracticality of Perfectly Secure Encryption)* If there is a perfectly secure scheme, then it must be the case that $|K| \geq |M|$, i.e., the key space is at least as big as the message space.

**Proof.** Let us assume that a perfectly secure encryption scheme yielded a ciphertext $c$ such that $c \leftarrow \text{Enc}(m, K)$. Given $c$, an adversary trying to break the scheme will compute all possible candidate plaintext values $m_i$ using all possible keys $K_i$ such that $m_i = \text{Dec}(c, K_i)$.

If we assume that the key space is smaller than the message space, i.e., if $|K| < |M|$, then there will not be complete mappings from key space $K$ to message space $M$. Therefore,

$$\Pr(\text{Enc}(m_i, K_i) = c) = 0 \quad \text{for some } m_i \in M \text{ and } \forall K_i$$

However, for the remaining messages $m_i$ such that $m_i = \text{Dec}(c, K_i)$, the probabilities are non-zero. Therefore, all messages are not equally likely to be the possible plaintext corresponding to a given ciphertext. This violates the property of a perfectly secure encryption (from Definition 2) and thus leads to a contradiction.

Since a perfectly secure encryption requires key space to be at least as big as the message space, it would not be possible to use a perfectly secure encryption scheme in practice. This motivates us to consider a weaker yet practical adversary – the one whose computation is bounded.

### 1.3 Polynomial Time Algorithms

**Definition 3** An algorithm which runs on an input of length $k$, is said to be a Polynomial Time Algorithm, if

$$T(k) = O(k^c)$$

for a constant $c$, where $T(k)$ is the time for computation.

### 1.4 Negligible Functions

**Definition 4** A function $f(k)$ is said to be negligible if $\forall c \geq 0, \exists k'$ such that

$$\forall k \geq k', \quad f(k) < \frac{1}{k^c}$$

2-4
A randomized polynomial time algorithm is called as a probabilistic polynomial time (PPT) algorithm.

See Figure 1.4 for an illustration of a polynomial time algorithm and a negligible function.

### 1.5 A realistic adversary

**Definition 5** A system that runs in polynomial amount of time $T(k)$ such that $T(k) = O(k^c)$ and whose probability of success $p(k)$ is negligible (represented as $\text{negl}(k)$)

**Example:** The DES encryption algorithm represented as $c = \text{DES}(K, m)$. Given a pair of plaintext and ciphertext, a brute force algorithm works as follows:

```latex
\text{for } i = 1, 2...q \\
\text{if } (c == \text{Enc} (K_i, m)) \\
\text{return } k_i
```

In this algorithm, the time $T(k)$ taken to learn the key, is expressed as:

$$T(k) = q.T_{\text{DES}}$$

If $q = k^2$ then $p(k) = \Pr(\text{success of probabilistic algo A})$

$$= 1 - \Pr(\text{failure of A})$$

$$= 1 - (1-p)(1-p)....(1-p) \text{ q times}$$

$$= 1 - (1 - p)^q$$

$$\geq q p \text{ (if } pq < 1)$$

where $p$ is the probability of success in 1 run.

### 1.6 Security Notions for Encryption Scheme in the above Realistic Adversarial Model

- **One Way Security** (*given* $c$, *learn* $m$)

**Example:** If a trader trades stocks such that he can either ‘Buy’ or ‘Sell’ (represented by 0 or 1), and if the cipher text $c$ is obtained by $\text{Enc}(0, k), \text{Enc}(1, k)\text{ etc}$

Even if the encryption scheme is one-way secure, i.e., an adversary can not learn the whole message $m$, given the ciphertext. However, the adversary may be able to learn some partial information about the message. For example, only one bit of the
Figure 1: poly(k); negl(k)
message (m) ‘0’ (sell) or sum of all the bits (how many stocks sold). This shows that one-way security is not a sufficiently strong security property. Of course, every encryption scheme should be one-way secure.